Intro to R for Epidemiologists

Lab 13 (4/16/15)

Part 0. Course evaluations

Please complete the evaluations for this course before preceding to the next section. Evaluations can be found at http://www.sph.emory.edu/rollins-life/evaluation/index.html

Part 1. Function to compute skewness

Skewness is a measure of asymmetry of a probability distribution. We have written the function below to compute skewness, which is defined by the following mathematical equation:

$$\frac{\frac{1}{n}(\sum_{i=1}^{n}(x_i - \bar{x})^3)}{sd(x)^3}$$

```
skewness <- function(x) {
    n <- length(x)
    sd1 <- sd(x)
    m <- mean(x)
    third.moment <- (1/n) * sum((x - m)^3)
    skew <- third.moment/(sd1^3)

    return(skew)
}
```

Modify the above function so that if the skewness is greater than 1 or less than -1, it prints a warning message that says, "Caution: Skewed data!" Use your new function to obtain the skewness of Petal length and Petal width by species in the Iris dataset as below.

```
skewness <- function(x) {
    n <- length(x)
    sd1 <- sd(x)
    m <- mean(x)
    third.moment <- (1/n) * sum((x - m)^3)
    skew <- third.moment/(sd1^3)

    if(skew>1 | skew< -1){
      warning("Caution: Skewed data!")
    }

    return(skew)
}

species <- group_by(iris, Species)
s_dat <- summarise_each(species,  funs(skewness), Petal.Length : Petal.Width)
s_dat
```

```
## Source: local data frame [3 x 3]
##
##      Species Petal.Length Petal.Width
## 1     setosa    0.1000954  1.17963278
## 2 versicolor   -0.5706024 -0.02933377
## 3  virginica    0.5169175 -0.12181190
```

Part 2. Debugging

The function posted on the website in `debugging.R` contains some errors. Copy the function into your R script and use debugging commands (e.g. `traceback()`, `debug(myfun)`, `browser()`) to find and fix the errors.

See function below for solution.

Part 3. Plotting

Use the error-free function `glmOR` from the Debugging section above and the `gfun` from class to create new nested function that takes only the arguments of the full dataset (e.g. `diabetes`), variables (e.g. `c("hdl", "chol")`), and outcome (e.g. `diab1`) and plots the corresponding univariate and multivariate odds ratios as shown below. Hint: you do not need to rewrite the `glmOR` function above, but only need to call it from within your new plotting function.

```
glmOR <- function(dat, vars, outcome) {

    # Set up outcome matrix
    or <- matrix(nrow = length(vars), ncol = 3)
    for(i in 1 : length(vars)) {
        # Run univariate regressions for each variable i
        coef1 <- glm(dat[, outcome] ~ dat[, vars[i]], data = dat,
                     family = "binomial")
        # Organize output
        lbub <- exp(confint(coef1)[2, ])
        or[i, ] <- c(exp(coef1$coef[2]), lbub)

    }
    # Add in variable names
    or <- data.frame(vars, or)
    # Add column names
    colnames(or) <- c("variable", "or", "lb", "ub")

    # Add type to univariate results
    or <- mutate(or, type = "univariate")

    # Get equation for multivariate results
    eqn <- paste(outcome, "~", paste(vars, collapse = " + "))
    # Run multivariate model
    coef2 <- glm(eval(eqn), data = dat, family = "binomial")
    lbub <- exp(confint(coef2)[-1, ])
    # Organize output
    mor <- data.frame(vars, cbind(exp(coef2$coef[-1]), lbub) )
    colnames(mor) <- c("variable", "or", "lb", "ub")
    # Add regression type
```

```
    mor <- mutate(mor, type = "multivariate")
    # browser()
    # Get all regression output
    output <- rbind(or, mor)

    return(output)
}




gfun <- function(data, vars, outcome) {
   glm1 <- glmOR(data, vars, outcome)

   size1 <- 18
   cols <- c("red", "blue")

   g1 <- ggplot(data = glm1, aes(x = type, y = or, colour = type)) +
        # Add ORs and confidence intervals
        geom_point(size = 3, shape = 20) +
        geom_errorbar(aes(ymin = lb, ymax = ub, colour = type),
           width = 0) +
        # Change plotting colors
        scale_color_manual(values = cols, name = "") +
        # Add axis labels
        ylab("Odds ratio") +
        xlab("") +
        # Add title
        ggtitle(paste("Covariates associated with", outcome)) +

        # Get rid of grey background
        theme_bw() +

        # Change size of labels
        theme(axis.text.y = element_text(size = size1),
              # Angle x axis labels
              axis.text.x = element_text(size = size1, angle = 20,
                hjust = 1,
              vjust = 1), legend.text = element_text(size = size1),
              axis.title = element_text(size = size1),
              # Remove legend
              legend.position = "none",
              plot.title = element_text(size = size1),
              strip.text = element_text(size = size1)) +
        # Add horizontal line at 1
        geom_hline(aes(yintercept = 1), colour = "grey50",
           linetype = "dashed")

    # Add faceting by type with free axes and 2 columns
    g1 + facet_wrap(~ variable, scales = "free", ncol = 2)
```

```
}
```

```
diab <- read.csv("diabetes.csv")

# load libraries
library(dplyr)
library(ggplot2)


gfun(diab, c("chol", "hdl"),  "diab1")
```